PA 926282

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

November 20, 2002

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: *60/418,634*
FILING DATE: *October 15, 2002*

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

M.K. Hawkins

M. K. HAWKINS
Certifying Officer

(5 10-16-02

60418634 . 101502

JC687 U.S. PTO
10/15/02

PROVISIONAL APPLICATION COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION under 37 CFR 1.53(b)(2).

Docket Number		Type a plus sign (+) inside this box	+
US020394P			

60/418634

INVENTOR(s) / APPLICANT(s)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
Li	Qiong		Cortlandt Manor, New York
Van der Schaar	Mihaela		Ossining, New York
Chen	Richard		Croton-on-Hudson, New York

TITLE OF THE INVENTION (280 characters max)

Error Recovery Method for Fine-Granularity-Scalability Coded Video Streaming

CORRESPONDENCE ADDRESS

Corporate Patent Counsel
U.S Philips Corporation
580 White Plains Road
Tarrytown, NY 10591

STATE	New York	ZIP CODE	10591	COUNTRY	U.S.A.

ENCLOSED APPLICATION PARTS (check all that apply)

- [X] Specification — Number of Pages: 14
- [ ] Small Entity Statement
- [ ] Drawing(s) — Number of Sheets: 0
- [ ] Other (specify)

METHOD OF PAYMENT (check one)

[ ]	A check or money order is enclosed to cover the Provisional filing fees	PROVISIONAL FILING FEE AMOUNT ($)	$150.00
[X]	The Commissioner is hereby authorized to charge filing fees and credit Deposit Account Number: 14-1270		

S:\hj\PX1SHJA0.NC0.doc

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No

☐ Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,

SIGNATURE [signature] Date: October 15, 2002

TYPED or PRINTED NAME: Dicran Halajian REGISTRATION NO. 39,703

☐ Additional inventors are being named on separately numbered sheets attached hereto

CERTIFICATE OF EXPRESS MAILING

Express Mail Mailing Label No. EL827840650

Date of Deposit: October 15, 2002

I hereby certify that this paper and/or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. 1.10 on the date indicated above and is addressed to the

Commissioner for Patents
Washington, D.C. 20231

Noemi Chapa — Noemi Chapa
Typed Name — Signature

This invention describes an ***error-recovery*** method for ***streaming Fine-Granularity-Scalability*** coded video over the Internet. In this method, error-recovery data, such as duplicated packets or redundant packets generated by FEC (Forward-Error-Correction) algorithms are separately stored in an error-protection track, along with original video tracks. When receiving error happens, the recovery is achieved by the receiver's subscription to the streaming channel that is assigned to deliver this protection track.

5. PRESENT STATE OF THE ART

Briefly describe the closest already-known technology that relates to the invention. This would include, for example, already existing products, methods or compositions which are known to you personally or through descriptions in publications.

Currently, there are two types methods used in practice for error-recovery: retransmission and forward-error-correction (or simply FEC). With retransmission, when errors occur at the receiver, such as a packet gets lost in the network, the receiver needs to first detect this loss event and then send a retransmission request back to source for recovery. With FEC, the source normally evenly embeds redundant packets into the original packet flows, hoping that when errors happen with original packets, some redundant packets can still be able to successfully make to the receiver, so the lost packets can be recovered from these redundant packets.

The advantage of retransmission is high bandwidth utilization. The disadvantage is retransmission involves quite a delay of recovery, at least a round-trip time, for example. Therefore, the retransmission method has limited usage, especially not feasible for interactive multimedia applications with real-time constraint, such as teleconferencing.

The FEC method is the opposite of the retransmission. FEC can quickly do error-recovery, but involves quite an overhead of bandwidth consumption. FEC is normally used when wireless links are parts of the communications channels.

6. ADVANCEMENT IN STATE OF THE ART

Briefly describe the unique advancement achieved by the invention. This may be done, for example, by describing a problem with the prior art that is solved or specific objects that are achieved by the invention.

In the past, there is a distinct line between these two methods. An application has to weigh to choose either retransmission or FEC. However, the Internet is heterogeneous and evolving. Same applications could operate in completely different network environments when started by different users widely spread across the Internet. Also, the network conditions are hard to predict, causing difficulty for choosing the right error-recovery methods once forever for all the applications.

An ideal solution for error-recovery would be that retransmission and FEC could be somehow combined together. An application has the freedom to dynamically choose either of them or combine in real-time according to its perceived network conditions.

Recently, we have seen a report that presents an error protection method that separate protection streams from media streams, and performs join/leave to achieve protection. This method is commonly referred to as adaptive FEC. This method has the following limitations:

- It uses IGMP (Internet Group Management Protocol) to signal the join/leave action, which may introduce a very long latency in the signaling process that eventually defeat the protection purpose, such as retransmission.
- It emphases on the FEC coding algorithm but lacks an architecture and protocols to carry out the goals of adaptive FEC.

Our invention describes a method that goes beyond just the basic idea of adaptive FEC. It presents a realistic architecture and specifies the protocols that are needed for carrying out protection. With our method, applications are able to switch between protection styles dynamically.

(ADD LINES AS NECESSARY, IF COMPLETING ON COMPUTER, OR ATTACH ADDITIONAL PAGES)

7. WHAT IS THE BEST WAY YOU KNOW OF TO IMPLEMENT THE INVENTION?

Briefly describe the invention and how it achieves the advancement described in paragraph 7.

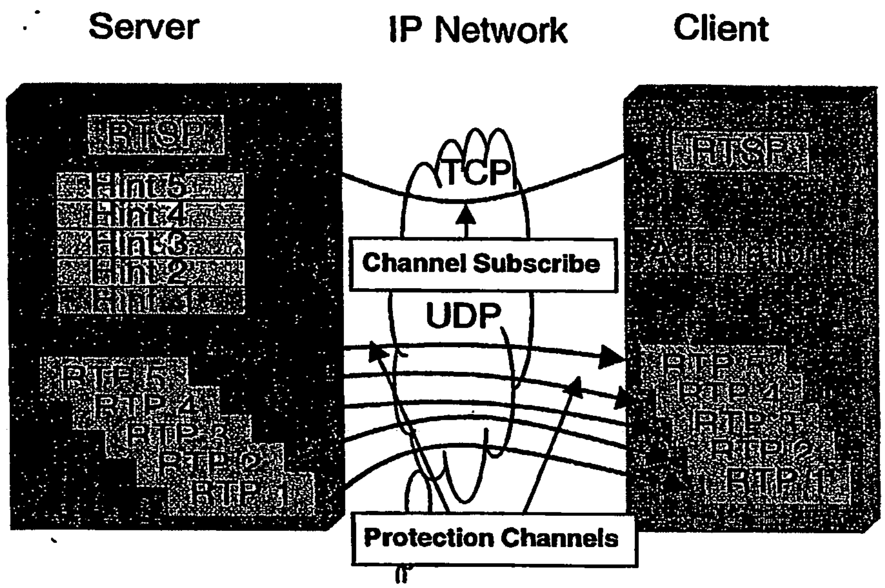
This error-recovery method can be seamlessly fitted into the overall FGS streaming architecture that was described by a previous disclosure (No. 702116). In this streaming architecture, FGS coded video are formatted into multiple tracks, and streamed to the clients using multiple network connections (or channels, in another words). One of these channels is reserved for the base layer (or track), while the rest for the enhancement layer (which is stored in multiple tracks).

Our error-recovery method can be implemented as the following.

- Add a separate protection track or tracks along with the original video tracks. The data of this track could simply be the duplications of the I-Frames of the base layer, or generated by a common FEC algorithm. The protection level is determined by the data stored in this protection track. In order for the server to use this protection track, a corresponding hinting track is also needed for this protection track.

- A signaling protocol for the subscription of the protection channel is needed, that can be done by using RTSP. The subscription to this protection channel could be permanent, behaving like the ordinary FEC method, or temporary, depending on perceived network loss conditions, or even to the extreme, one decodable unit at a time to mimic the retransmission method.
- The receiver needs to monitor its receiving quality and actively triggers the protection channel when it deems necessary.

This method is also schematically shown in the following Figure.



In this figure, one or more RTP channels will be used for protection. Protection channel subscriptions will be done using RTSP, and initiated by the client.

The advantage of this method may reflect in the following.

- Fully take advantage of the QuickTime file format, and the popular Darwin Streaming server architecture.
- Protection data are separated from protected data. Change the protection data can change the protection level, but the protection method remains the same.
- With this method, applications can dynamically choose either retransmission-like protection or FEC-like protection, or in the middle, therefore, gaining better protection performance.
- Using RTSP instead of IGMP can achieve faster protection and provide more flexibility to applications.
- This method works well for both unicast and multicast.

(ADD LINES AS NECESSARY, IF COMPLETING ON COMPUTER, OR ATTACH ADDITIONAL PAGES)

*********<u>PLEASE NOTE</u>: IF WE DECIDE TO FILE AN APPLICATION ON THIS INVENTION, THE ATTORNEY WRITING THE APPLICATION WILL NEED THIS INFORMATION FROM YOU IN AS MUCH <u>DETAIL</u> AS POSSIBLE IN ORDER TO COMPLETE THE APPLICATION.**

8. <u>DISCLOSURE OUTSIDE OF PHILIPS</u>

If the invention has been or will be disclosed publicly or to anyone other than a Philips' employee,

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 M8944
Shanghai, October 2002

Title: A Flexible Streaming Architecture for Efficient Scalable Coded Video Transmission over IP Networks

Source: Qiong Li, Mihaela van der Schaar (Philips Research)

Subgroup: System and Video

Purpose: Discussion

Abstract

In this contribution we present a streaming architecture framework for streaming scalable coded video over IP networks. This architecture uses multiple channels (or more precisely, multiple IP connections for unicast and multicast-group channels for multicast) to deliver scalable coded video. This framework includes hinting scheme for scalable video, a protection strategy for achieving unequal protection and protection on demand, and extensions to standard Internet streaming protocols (RTSP, SDP) for conducting channel controls. The concept of this proposed framework has been verified by a prototyping implementation.

1. Introduction

With the rapid development of broadband Internet technologies, video streaming is envisioned to become the dominant Internet application in the near future. Similarly, the falling cost of WLAN products has also led to their increased use in consumer homes, and although currently most WLANs are predominantly used for data transfer, the higher bandwidth provided by new WLANs technologies such as IEEE 802.11a and IEEE 802.11g will ultimately lead to their increasing use for video transmission. Furthermore, future wireless video applications will have to work over an open, layered, Internet-style network with a wired backbone and wireless extensions. Therefore, common protocols will have to be used for the transmission across both the wired and wireless portions of the network. These protocols will most likely be future extensions of the existing protocols that are based on the Internet Protocol (IP).

Consequently, due to the inherent resource sharing nature of the Internet and wireless networks, multimedia communications of the future will mainly use variable bandwidth channels. Hence, if streaming of video content is performed over this type of networks, the instantaneous data rate must frequently be tailored to fit the available resources. This can be achieved in a very flexible way by the approach of scalable coding. Scalable video-coding schemes are able to provide a simple and flexible framework for transmission over heterogeneous networks, since:

- They enable a streaming server to perform *minimal real-time* processing and rate control when outputting a very large number of simultaneous unicast (on-demand) streams (unlike transcoding or simulcast approaches).

[redacted]

- They are highly adaptable to unpredictable bandwidth variations due to heterogeneous access-technologies of the receivers (e.g. analog modem, cable mode, xDSL, etc.) or due to dynamic changes in network conditions (e.g. congestion events).
- They enable low-complexity decoding and low-memory requirements to provide common receivers (e.g. set-top-boxes and digital televisions), in addition to powerful computers, the opportunity to stream and decode any desired video content.
- They are able to support both multicast and unicast applications. This, in general, eliminates the need for coding content in different formats to serve different types of applications. Moreover, for multicast applications, the scalable coded streams require less bandwidth for transmission.
- They are resilient to packet and bit-error losses, which are quite common over the Internet and wireless networks.

Examples of scalable coding schemes are the MPEG-4 Fine Granularity Scalability (FGS), Advanced FGS, Data-Partitioning, MPEG-4 Spatial and Temporal Scalabilities, emerging Motion-Compensated Wavelet Solutions etc. However, in order to provide the required adaptation to bandwidth variations, device characteristics and user requirements, the scalable video coding needs to be transmitted using an appropriate streaming architecture.

The MPEG-4 Systems Group has developed and standardized the streaming strategy for non-scalable coded video over IP networks. However, a novel streaming architecture is required for the transmission of scalable video formats standardized by MPEG that is able to take advantage of their flexibility in order to efficiently adapt to channel conditions, complexity constraints and user preferences.

In this contribution, we describe such a flexible streaming architecture framework that includes a pre-processing method (hinting) for flexible scalable video packetization, an efficient protection strategy and the necessary streaming control protocols. Particularly, we propose a multi-track-hinting concept for scalable video, and extend the functions of standard Internet streaming protocols (RTSP, SDP) to enable flexible adaptation. One principle we have followed in the process of defining this framework is that the scalable video streaming system architecture should be compatible with that of the non-scalable streaming system defined by MPEG, such that a general purpose MPEG-4 streaming server can deliver both types of video.

2. Multi-track Hinting

Multi-track hinting is a method proposed in this contribution for structuring layered video into a file format that is backwards compatible with the MPEG-4 media file format standard, therefore making it possible to use a general-purpose MPEG-4 streaming server to stream layered video. Using this method to prepare layered video for streaming will enable the server (without major modifications) to automatically use multiple channels to deliver the scalable video over IP networks. The channels can be controlled using the mechanisms described in the next section.

The MPEG-4 standardization body has developed a standard media file format (.mp4) [1] that contains timed media information for multimedia presentation either locally or remotely (such as streaming). This format is deliberately designed with high flexibility and extensibility in order to facilitate interchange, management, editing, and presentation of the media.

The standard file format has an inherent hierarchical structure. The basic building blocks used in the construction of mp4 files are called Boxes. A box is a specially designed data structure that contains a certain type of media data. Each box has a type name, reflecting the type of data it contained. Also, a box can contain other boxes, and so on so forth to form a hierarchical structure.

The general structure of a mp4 file format for streaming is illustrated in Figure 1. Normally, an mp4 file starts with a root box called moov. The moov box further contains other boxes, such as, boxes for storing elementary bit steams, boxes for storing synchronization information (or called movie tracks), and boxes for storing hints used by streaming server to generate packets out of the elementary bit streams (these boxes are called hint tracks). On the highest level of abstraction, a mp4 file can be view as a structure containing

elementary bit streams generated by encoders, movie tracks to guide player for local playback, and hint tracks for streaming the media over packet-based networks.

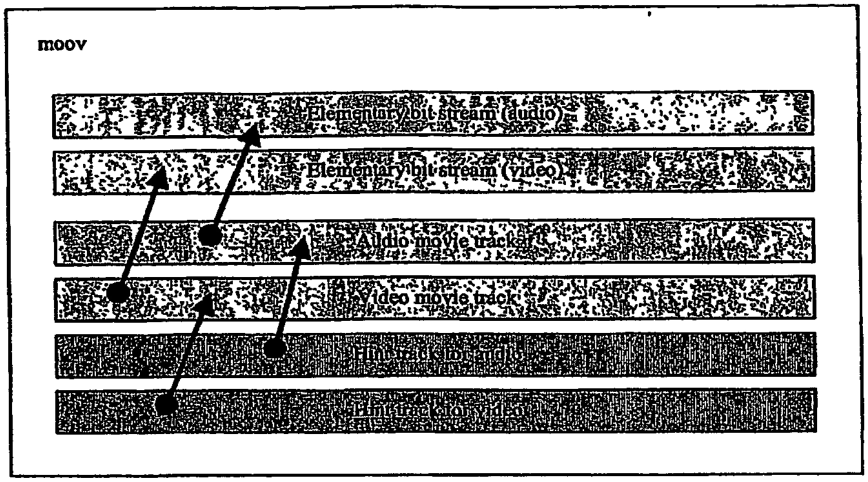


Figure 1: mp4 file format

The arrows in Figure 1 indicate that the movie tracks are related to elementary streams, and the hint tracks to the movie tracks. The movie tracks contain information (timing and data pointers) that a player will use to extract the corresponding media data for presentation at the designated time. Hint tracks contain information (such as timing, data pointers and data for packet headers) that a server will use to generate packets out of elementary bit streams. Hint tracks are created by a tool/processor called "hinter" based on movie tracks.

When mp4 file is used in a streaming application, normally the server will establish a number of RTP connections equal to the number of hint tracks that are contained in the file. There is a one-to-one relationship between an RTP connection and a hint track. Each RTP connection will be assigned with a hint track, and is responsible for delivering packets generated from that track.

However, the mp4 file format described above does not explicitly address the requirement of layered video streaming. In layered video coding, compressed video is structured into multiple layers. These layers can be added up progressively to improve video quality. To take advantage of the scalability provided with layered coding, normally multiple RTP connections will be used for layered video streaming, and the totally number of the connections will be adapted to the network conditions.

Layered video coding normally generates one elementary bitstream that can be divided in sub-layers having different priorities/importance. If we simply apply the generic mp4 file format to the layered video elementary streams - in which for each elementary stream the authoring tool generates a movie track and then further generates a hint track - we will end up with a scheme that can only use one RTP connections to stream the layered video. Obviously, scalable coding based on this inflexible streaming strategy does not allow for the desired adaptation to channel characteristics, complexity etc.

In order to use a general-purpose streaming server designed to stream generic mp4 files for efficient layered video streaming, and at the same time to be able to set up multiple RTP connections for streaming corresponding to the various importance sub-layers, we propose a hinting method that can generate multiple hint tracks out of a single movie tracks. This method can generate files that still comply with the mp4 file format. When a single movie track is hinted using multiple hinting tracks, the elementary stream pointed by the movie track will be delivered over the network by multiple RTP connections, thereby providing the streaming system the flexibility to adapt to network conditions by adjusting the transmitted number of scalable layers.

The application of the multi-track hinting concept is beyond the layered coding case. For example, this hinting method can be successfully applied to a video stream to associate different types of video frames (I-, P-, B-frames) to different hint tracks. In this way, the temporal video scalability is easily achieved by the streaming system.

The multi-track hinting can also be viewed as an extension to the generic mp4 file format.

The basic idea of multi-track hinting is illustrated in Figure 2. As shown in Figure 2, multiple hint tracks are generated for a single movie track, therefore multiple RTP connections will be used by the server to deliver the elementary stream pointed by this movie track. The multi-to-one relationship between hint track and movie track breaks the original one-to-one relationship employed for hinting non-scalable streams, thereby providing the flexibility necessary for scalable video transmission.

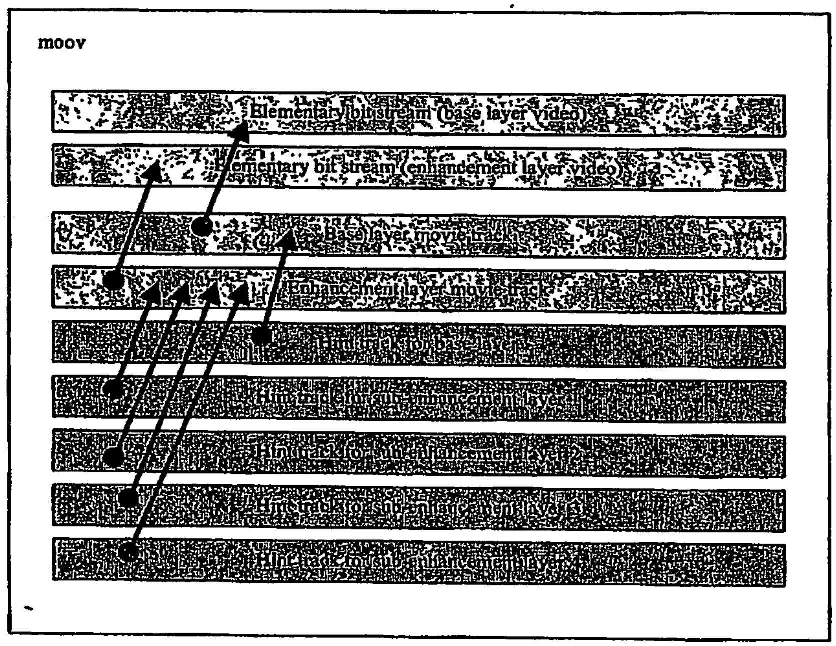


Figure 2: Proposed multi-track hinting file format

Using this multi-track hinting method, the elementary stream is only virtually divided into multiple sub-streams, therefore, the physically unchanged elementary stream stays the same and can be played locally as before.

For illustration, we show a pseudo code that describes the algorithm for MPEG-4 FGS multi-track hinting in Figure 3. In Figure 3, the function max_channel_allocation(i) will determine the bit rate that will be allocated to the *i*th RTP connection associated with the *i*th hinting track. Therefore, the bit rates of the streaming channels are pre-determined at the hinting stage.

```
while ((sample_size=get_next_fgs_sample()) > 0)  { //loop through all samples
        remaining = sample_size;
        offset = 0;
        for (i = 0, i < max_hint_track_num, i++) { //loop through all hinting tracks
                if( remaining > 0) {
                        channel_remainning = remaining > max_channel_allocation(i) ?
                                        max_channel_allocation(i) : remaining;
                remaining = remaining > channel_remaining ?
                                        remaining – channel_remaining : 0;
                while (channel_remaining > 0) { //generate packets
                        if ( channel_remaining <= max_packet_payload_size) {
                                packet_length = channel_remaining;
                        } else {
                                packet_length =  max_packet_payload_size;
                        }

                        add_packet_to_hint_track_[i](offset);
                        offset += packet_length;
                        channel_remaining -= packet_length;
                }
        }
}
```

Figure 3: Pseudo code of proposed FGS multi-track hinting algorithm

It is possible to develop algorithms for packetization and rate-allocation optimizations when specific network conditions and codec characteristics are taken into account. These algorithms are application specific, and will not be further discussed in this contribution.

However, it is worth pointing out that, for a single elementary bit-stream, there could be more than one layering schemes, each reflecting a packetization strategy targeted to a specific network condition. In multi-track hinting, each of these layer schemes can generate a set of hinting tracks. All hinting tracks associated with different layering schemes can be stored in the same file, and made know to users via SDP channel profiling mechanism which is discussed in next section, so that users can have options to choose among different layering schemes when join a streaming session.

3. SDP Extension

The SDP (Session Description Protocol) is generally used by a streaming server to describe a streaming session to a client. In the case of layered video streaming, a few session features that are unique to layered video streaming that have not been expected by the original design of the SDP specified in RFC 2327. These features are:

- The video is not just a single stream, but rather contains multiple layer streams transmitted over different IP connections (or channels) with different priorities.
- The server could recommend different possible channel combinations (referred to as channel profiling) to a client based on network conditions or client capabilities, or offer multiple channel profiling options for clients to choose. For example, some clients may prefer low frame rates but high quality images, while others may prefer the opposite. These different preferences can be satisfied using the channel profiling feature.

- Different layers could be protected independently by protection tracks (see detail discussion in Section: On-Demand Protection). A server needs a mechanism to signal to the clients the availability of these protection tracks.

We propose the following extension to SDP:

Session Level:

Syntax:

a=pname: <channel_profile_name/min_bandwidth – max_bandwidth>

Explanation:

"pname": indicates that this attribute is to define a channel profile.
"channel_profile_name": defines the name of this profile.
"min_bandwidth – max_bandwidth: defines the bandwidth range that required for fully receiving all channels in this profile.

Example:

"a=pname:basic/56 – 1000" specifies that there is a profile called "basic" and its required bandwidth range is between "56 – 1000" kbps.

Media Level:

Syntax:

a=profile:<profile_name/track_priority>

Explanation:

"profile": indicates that this attribute is to tell which channel profile this media belongs to.
"profile_name": specifies the profile name.
"track_priority": specifies the priority level of this media (or track) in this particular profile.

Example:

"a=profile: basic/1" specifies that this media appears in the channel profile named as "basic" with an assigned priority level of "1." In media level, there could be multiple "a=profile: …" lines with different profile name under the same media.

Syntax:

a=protection: <trackID=value>

Explanation:

"protection": indicates that this media (or track) is protected by another track, such as a track contains FEC data of this media.
"trackID=value": specifies the track control ID of the protection track, or the ID of the hinting track of the protection track.

Example:

"a=protection:trackID=12", specifies that this media, under which this attribute line appears, is protected by a protection track that is hinted and the hinting track ID is "12".

With these extensions, a streaming server will be able to properly describe a layered streaming session to a client, and the client can selective subscribe to a group of channels defined by the channel profile it chooses from the offered options.

4. Channel Control Using RTSP

The fundamental advantage of layered video streaming is that video is transmitted through multiple channels (or RTP channels). The more channels are used, the higher the network bandwidth consumed. The streaming system needs a mechanism to start and stop these channels dynamically depending on network conditions, such as available bandwidth. We have prototyped an adaptive streaming system, in which we extended the RTSP (RFC 2326) function for rate control. The rate adaptation is achieved through channel subscription and un-subscription, in the following way:

- When video layers and protection layers are loaded by the server, it associates each layer with an *activity* status variable. A layer's status could be either ACTIVE or INACTIVE. The server will only deliver layers that are in the ACTIVE status to the client.
- RTSP's "SET PARAMETER" command is used by the client to set the *active* status variable of each layer in real-time.
- The client estimates network conditions and decides when to start or stop a layer.

This scheme is similar to the receiver-driven layered multicast [2] in the sense that in both schemes the receiving side actively performs the adaptation. The difference is that in our scheme the RTSP is used instead of IGMP to avoid long IGMP latencies that could affect the adaptation efficiency. However, for our scheme to support multicast and suppress the implosion of RTSP control message to a busy server, strategically located RTSP proxies in large scale networks to filter and merge RTSP control messages before they reach the server are necessary, which is beyond the scope of this contribution.

The RTSP command used for channel control is illustrated in the following example:

```
C→S:  SET_PARAMETER rtsp://130.140.67.83/sample.mp4 RTSP/1.0
      CSeq: 11
      Session: 3453643
      Content-length: 13
      Content-type: text/bool

      Track_04: 1        // the 4th track is set to be 1 (ACTIVE)

S→C:  RTSP/1.0 200 OK
      CSeq: 11
      Date: 28 Jan 2002 15:32:10 GMT
```

The protection layers (or tracks) are controlled in the same way as the video layers. Nevertheless, the client will engage different criteria to invoke or suppress the protection channels, depending on the loss characteristics and/or desired protection levels that can be determined by each client individually (based on e.g. employed buffer size, concealment strategy).

The general procedures of layered streaming is illustrated in Figure 4 using FGS as an example.

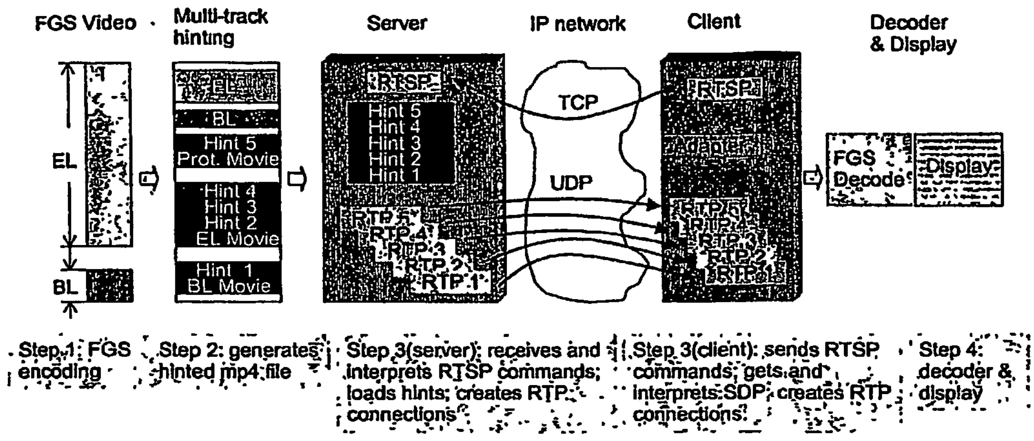


Figure 4: General streaming procedures.

5. On-demand Protection

In most streaming systems, mainly two types of error-recovery methods are used: retransmission and forward-error-correction (or simply FEC). The retransmission method has the advantage of high bandwidth utilization, but the disadvantage of long recovery delays that may not be tolerable for applications having strict delay constraints. The FEC method is the opposite of the retransmission. FEC can quickly do error-recovery, but involves an overhead of bandwidth consumption.

In the past, there is a distinct line between these two methods. An application design chooses either retransmission or FEC. However, the IP-based networks are heterogeneous and evolving. Applications could operate in completely different network environments, making the network conditions hard to predict. This situation leaves application designs with difficulties for choosing the right error-recovery methods for all possible operating scenarios.

An ideal solution for error-recovery would be that retransmission and FEC could be somehow combined together, and the application has the freedom to dynamically choose either of them or combine them in real-time according to its perceived network conditions.

The hybrid ARQ and the adaptive FEC reported in [3] [4] are attractive solutions that combine the strengths of both retransmission and FEC. In the hybrid ARQ, the video data are pre-encoded by some FEC coding scheme, such as Reed-Solomon (RS) codes, and then the sender and receiver use a specially designed ARQ-like protocol to perform the protection.

In the adaptive FEC [4], FEC data are separated from original media streams, and "join"/"leave" commands are employed to achieve adaptive protection. This method has the following limitations:

- It uses IGMP (Internet Group Management Protocol) to signal the join/leave action, which may introduce a very long latency in the signaling process that eventually defeat the protection purpose, such as retransmission.
- It emphases on the FEC coding algorithm but lacks an architecture and protocols to carry out the goals of adaptive FEC.

The scheme described in this contribution goes beyond just the basic idea of adaptive FEC. It presents a realistic architecture and specifies the protocols that are needed for carrying out adaptive and efficient protection. With our scheme, applications are able to switch between different protection strategies dynamically.

Our error-recovery method can be implemented as follows.

- A separate protection track or tracks is/are added along with the original video tracks. The data of this track could simply be the duplications of the I-Frames of the base layer, or be generated by a common FEC algorithm. The protection level is determined by the data stored in this protection track. In order for the server to use this protection track, a corresponding hinting track is also needed for this protection track.
- A signaling protocol for the subscription to the protection channel is needed, that can be done by using RTSP. The subscription to this protection channel could be permanent, behaving like the ordinary FEC method, or temporary, depending on perceived network loss conditions, or even to the extreme, one decodable unit at a time to mimic the retransmission method.
- The receiver needs to monitor its receiving quality and actively triggers the protection channel when it deems necessary.

Figure 5 shows how protection tracks are structured and stored in an MPEG-4 file.

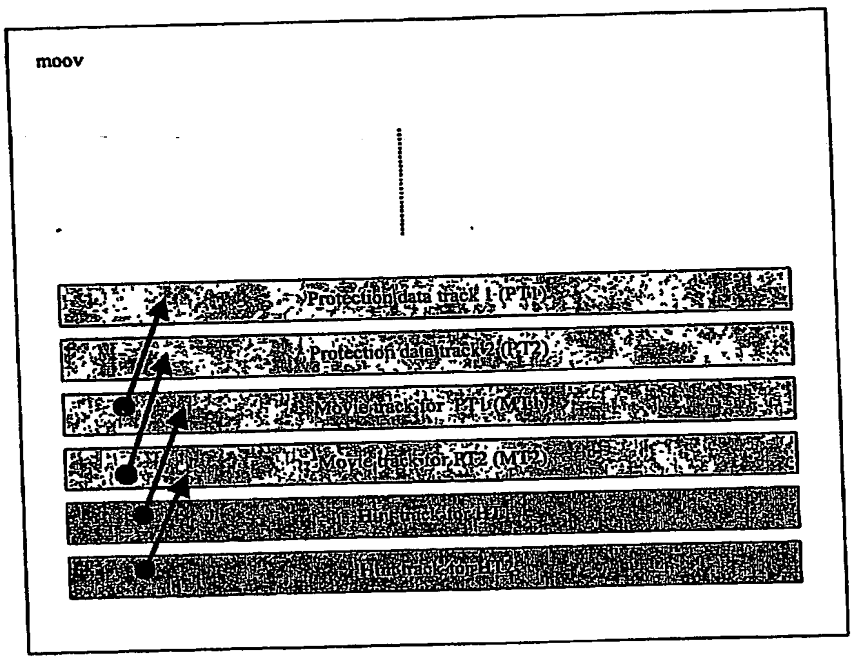


Figure 5: Proposed arrangement of protection tracks in an MPEG-4 file.

In Figure 5, PT1 and PT2 contain the FEC data of some video tracks. MT1 and MT2 allow these protection tracks to be used locally, if needed. HT1 and HT2 make these protection tracks available to remote users through streaming.

The operation of this error-recovery method is schematically shown in Figure 6.

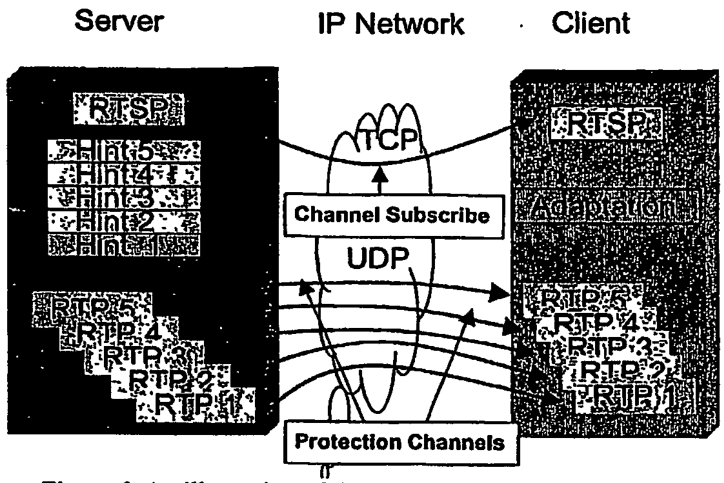


Figure 6: An illustration of the error-recovery scheme.

In the example shown in Figure 6, channel RTP1 – RTP3 carry the video data of different layers. RTP4 and 5 carry the protection data. Both video channels and protection channels are controlled using RTSP in the same way, the client determines when it wants to turn on/off a particular channel.

The RTSP command used for protection control is shown as the following.

C→S: SET_PARAMETER rtsp://130.140.67.83/sample.mp4 RTSP/1.0
CSeq: 32
Session: 3453643
Content-length: 35
Content-type: text/bool/integer

Track_11: 1 // the 11th track is set to be 1 (ACTIVE)
range: 34521 - 34570 //50 packets are requested

S→C: RTSP/1.0 200 OK
CSeq: 32
Date: 28 Jan 2002 15:33:10 GMT

Explanation:

Parameter *range* Syntax:

range: start_seq_num - end_seq_num

Implication:
- as **FEC**: when end_seq_num = ∞
- as **Retransmission**: when end_seq_num = start_seq_num + 1
- as **Hybrid**: for other cases

6. Conclusions

The advantage of this framework for adaptive and efficient video streaming is highlighted below.

- It takes full advantage of the MPEG-4 file format, and a general-purpose MPEG-4 server can perform adaptive protection to streaming applications.
- Protection data are separated from protected data and changing the protection data can change the protection level or strategy, but the protection procedures remains the same.
- With this method, applications can dynamically choose either retransmission-like protection or FEC-like protection, or hybrid ARQ, therefore, gaining better protection performance.
- Using RTSP instead of IGMP can achieve faster protection and provide more flexibility to applications.
- This streaming method is efficient for both unicast and multicast.

This contribution only outlines an architecture framework for layered video streaming. We hope this work can bring up interesting discussions regarding layered video streaming in MPEG-4 community. Our prototyping work demonstrated that the multi-hinting method, RTSP channel control as well as the on-demand protection concepts briefly described in this contribution can effectively be used for Internet and in-home wireless transmission of video. Future work includes refining the syntax and semantics design of involved protocols and adaptation algorithms.

Reference:

[1] D. Singer, W. Belknap, G. Franceschini, "ISO Media File Format Specification — MP4 Technology under consideration for ISO/IEC 14496-1:2002 Amd 3," Committee Draft, ISO/IEC JTC1/SC29/WG11 MPEG01/N4270-1, July 2001.

[2] S. McCanne, V. Jacobson, M. Vetterli, "Receiver-driven Layered Multicast," Proceedings of ACM SIGCOMM'96, pp117 – 130, 1996.

[3] Majumdar, D. Grobe Sachs, I. V. Kozintsev, K. Ramchandran "Multicast and Unicast Real-Time Video Streaming Over Wireless LANs", IEEE Transactions on Circuits and Systems for Video Technology, Vol.12, No.6, pp524 – 534, June 2002

[4] P. Chou, A. E. Mohr, R. Wang, S. Mehrotra, "FEC and Pseudo ARQ for Receiver-driven Layered Multicast of Audio and Video", Microsoft Research Technical Report MSR-TR-99-86, 1999.

604[illegible]634.101502

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

QIONG LI ET AL

Serial No.

Filed: CONCURRENTLY

Atty. Docket

US020394P

Title: ERROR RECOVERY METHOD FOR FINE-GRANULARITY-SCALABILITY CODED VIDEO STREAMING

Commissioner for Patents
Washington, D.C. 20231

APPOINTMENT OF ASSOCIATES

Sir:

The undersigned Attorney of Record hereby revokes all prior appointments (if any) of Associate Attorney(s) or Agent(s) in the above-captioned case and appoints:

DICRAN HALAJIAN **(Registration No. 39,703)**
c/o U.S. PHILIPS CORPORATION, Intellectual Property Department, 580 White Plains Road, Tarrytown, New York 10591, his Associate Attorney(s)/Agent(s) with all the usual powers to prosecute the above-identified application and any division or continuation thereof, to make alterations and amendments therein, and to transact all business in the Patent and Trademark Office connected therewith.

ALL CORRESPONDENCE CONCERNING THIS APPLICATION AND THE LETTERS PATENT WHEN GRANTED SHOULD BE ADDRESSED TO THE UNDERSIGNED ATTORNEY OF RECORD.

Respectfully,

Michael E. Marion, Reg. 32,266
Attorney of Record

Dated at Tarrytown, New York
this 15th day of October, 2002.

S:\hj\MX15HJC0 NC0.DOC